

TRI-004 US

PATENT

An Input/Output Circuit With User Programmable Functions

Brian Fox
Andreas Papaliolis
Steven P. Winegarden
Edmond Y. Cheung

Express Mail
Date of Deposit: 4/4/03
I hereby state that I am causing this paper or fee to be deposited with the United States Postal Service Express Mail Post Office to Addressed service on the date indicated above and that this paper or fee has been addressed to the Commissioner for Patents, PO Box 1450, Alexandria, Virginia 22313-1450
Christopher P. Marshall
(Signature of person mailing paper or fee) 3/16/2003
(Date signed)

FIELD OF THE INVENTION

The present invention relates to integrated circuits, such as configurable system logic devices and configurable system-on-chip products. More specifically, the present invention relates to a method and structure to provide an input/output circuit with user programmable functions.

DISCUSSION OF RELATED ART

The input/output (I/O) circuit of a conventional integrated circuit (IC) acts as an interface between the integrated circuit and the outside world. Conventional ICs have pre-determined internal wired connections. In general, the signals passed through the I/O circuit of a conventional non-programmable IC are pre-defined and come from hardwired locations. As a result, the I/O pins of the IC have known state requirements. For example, a particular I/O pin may need to be grounded when it is not receiving a signal from an external source. Typical circuitry within a conventional I/O circuit includes buffers (input and output) and registers (for input, output, and output enable signals). I/O circuit design attempts to optimize switching speed while minimizing switching noise. The switching speed of a signal, typically a clock signal, is the time it takes to transition from one logic state to another. The amount of switching noise produced is in part attributable to the

1 amount of overshoot occurring in the transition of the
2 clock signal from one logic state to another. I/O circuit
3 register design attempts to optimize setup and hold times
4 of the register relative to a known clock signal. The
5 setup time of a register is the amount of time prior to a
6 controlling clock edge during which a data signal must not
7 change. The hold time of a register is the amount of time
8 after a controlling clock edge during which a data signal
9 must not change. If a data signal changes during the setup
10 time or hold time, the signal at the output of the register
11 is unpredictable.

12 A short setup time and a zero or negative hold time
13 relative to a clock signal are very desirable. However, as
14 the hold time relative to a clock signal becomes more
15 negative, the setup time relative to the same clock signal
16 becomes larger. Therefore, to conventionally optimize hold
17 times, the delay on the data input of a register relative
18 to a clock signal is carefully simulated to make the hold
19 time as near zero as possible.

20 Programmable logic, such as Field Programmable Logic
21 devices (FPLD) and Configurable System Logic (CSL) resident
22 on Configurable Systems on a Chip (CSoC), provide built-in
23 circuits that can be programmably interconnected, thus
24 allowing a user to implement different designs "in the
25 field" using the device. Typically, these designs are
26 implemented by using Computer-Aided Design (CAD) "Place and
27 Route" software. The CAD Place and Route software
28 determines the placement of the designed circuits on the
29 CSL and programs the memory elements that control the
30 interconnections of the designed circuits.

31 Additional requirements beyond those of conventional
32 non-programmable integrated circuits are needed due to the

1 programmable nature of the CSL. For example, the
2 arrangements of designed circuits on the CSL must
3 facilitate implementation of useful functions by CAD
4 software. This facilitation is typically accomplished by
5 providing a wide selection of functional blocks and routing
6 resources and providing a programmable means to connect
7 both blocks and routing. Unfortunately, more flexible
8 programmability of the CSL causes more complex CSL
9 production testing procedures.

10 In production testing, the CSL must be programmed a
11 large number of times in differing configurations to
12 exhaust the combinations of possible interconnections of
13 built-in circuits. As the flexibility of programming the
14 CSL increases, the number of possible combinations of
15 interconnections of built-in circuits increases. The cost
16 of production of the CSL increases with the increase in
17 complexity of CSL testing procedures.

18 During programming of the configuration memory
19 elements, the internal logic of the CSL is unstable and
20 unpredictable. The internal signals from this logic may be
21 provided to output buffers and be driven to off-chip
22 components. Therefore, a CSL requires a means to generate
23 predictable states in the programmable I/O circuitry (PIO)
24 of the CSL. The PIO performs the I/O function of the CSL.

25 Conventionally, using a tri-state buffer in an IOB
26 while connecting the output pad of the IOB to a "weak pull-
27 up" circuit is adequate in most situations. A weak pull-up
28 circuit connected to an output pad provides a connection to
29 a logic one that can be easily overcome by a signal
30 asserted on the output pad. For example, Xilinx Inc.
31 provides a weak pull-up circuit and a tri-state buffer in
32 their IOB shown on page 4-25 of the Xilinx Programmable

1 Logic Data Book, Version 1.03 (July 30, 1996). A
2 conventional pull-up circuit as described above limits the
3 flexibility of a system designer by limiting the options
4 available for defining a given logic state on the output
5 pad.

6 Some IOBs within conventional FPGAs allow input
7 signals to pass directly into selected routing channels.
8 However, the number of channels available for a direct
9 connection to a input terminal providing input signals is
10 seriously limited. Other IOBs have latched or registered
11 input signals before routing the signals into other routing
12 channels. However, this latching or registering
13 significantly increases the circuit area as well as the
14 delay involved in selecting signals for routing.

15 Conventional programmable logic architectures
16 comprising homogeneous arrays of smaller tiles commonly
17 utilize a unique design and layout at each edge of the tile
18 array to control the I/O interfaces between the logic array
19 and external signals. Although the function and
20 connectivity of the tile at each edge may be individually
21 customized to take particular advantage of the location of
22 the tile, the expense of this customization is greatly
23 increased effort for design, verification, and layout.

24 Therefore, a need arises for an improved IOB which
25 increases the number of channels available for direct
26 connection to an input terminal while reducing circuit area
27 and routing delay. Further, a need arises for an I/O block
28 tile that makes externals driven or received by the tile
29 similarly available to internal tiles without regard to the
30 edge location of the tile.

1 SUMMARY OF THE INVENTION

2 The I/O circuit of the present invention provides
3 optimal flexibility and performance using a number of
4 different structures and methods.

5 The present invention provides a combination control
6 circuit for an input pad wherein the signal on the pad may
7 be pulled up to a logic one, pulled down to a logic low, or
8 pulled to the logic value present on the input pad.

9 The present invention uses a configuration signal to
10 select a value for an enable signal to either disable an
11 output buffer during configuration, or to enable the output
12 buffer according to a programmed value of a memory element.
13 If the output buffer is enabled during configuration, the
14 driven value can be programmed by memory elements. If the
15 output buffer is disabled during configuration, the output
16 pad can be pulled up to a logic one or pulled down to a
17 logic zero based on a logical function of programmed memory
18 elements. Thus, the present invention guarantees
19 predictable output characteristics when a configurable
20 system logic device is being programmed.

21 The present invention further provides a delay circuit
22 that programmably varies the amount of the delay through
23 the circuit. Specifically, the present invention provides
24 a signal propagation delay from a programmable input/output
25 (PIO) to an internal routing structure. As a result, zero
26 hold time for an arbitrary input register relative to a
27 fixed global clock is achieved.

28 In accordance with another aspect of the present
29 invention, an OR gate combines the inputs from a horizontal
30 routing channel and a vertical routing channel and provides
31 the combined signal to the data input of an output
32 register. This use of the OR gate allows one test

1 configuration during production testing to test two input
2 signals into the register. In contrast, conventionally,
3 two test configurations would be required to perform these
4 two tests. Thus, the invention significantly reduces
5 product test time in production of a CSoC.

6 In yet another aspect of the present invention, a
7 transistor is coupled to two PIO input pads. As a result,
8 the input pads may be coupled together by controlling the
9 voltage at the gate of the transistor. This direct
10 coupling allows fast signal transfer between the input
11 pads.

12 The present invention also addresses flexible routing
13 structures. In one embodiment, a bypass latch is used to
14 enable system routing to receive both a current input
15 signal (through a first channel) and the last value of the
16 input signal (through a second channel). The latch may be
17 disabled, thereby allowing the routing to receive the
18 current input signal via both channels.

19 In a final aspect of the present invention, a
20 plurality of identical input/output block tiles are
21 provided, thereby ensuring that each interior logic tile
22 coupled to an IOB tile receives the same signal set,
23 regardless of the edge to which the IOB tile is coupled.
24

25 BRIEF DESCRIPTION OF THE DRAWINGS

26 Figure 1 is a block diagram of a conventional field
27 programmable logic device;

28 Figure 2 is a schematic diagram of a conventional IOB
29 for the field programmable logic device of Figure 1;

30 Figure 3 is a schematic diagram of a configurable
31 system on a chip in accordance with an embodiment of the
32 present invention;

1 Figure 4 is a schematic diagram of a PIO in accordance
2 with an embodiment of the present invention;

3 Figure 5 is a schematic diagram of another PIO in
4 accordance with another embodiment of the present
5 invention;

6 Figure 5A is a table describing a mode of operation of
7 the PIO of Figure 5;

8 Figure 5B is a table describing another mode of
9 operation of the PIO of Figure 5;

10 Figure 6 is a schematic diagram of a programmable
11 delay circuit in accordance with an embodiment of the
12 present invention;

13 Figure 7 is another embodiment of the input delay
14 circuit of Figure 6;

15 Figure 8A is a schematic diagram of placement-
16 independent edge tiles in a semi-homogeneous logic array in
17 accordance with an embodiment of the present invention;

18 Figure 8B is a detailed schematic diagram of an edge
19 tile 850 of Figure 8A showing the multiplexers at the O and
20 E terminals of PIO 400 (Figure 4);

21 Figure 9 is a schematic diagram of fast switches in
22 accordance with an embodiment of the present invention; and

23 Figure 10 is a schematic diagram of a latch bypass in
24 accordance with an embodiment of the present invention.

25

26 DETAILED DESCRIPTION OF THE INVENTION

27 Figure 1 is a block diagram of a conventional field
28 programmable logic device (FPLD) 101 including IOBs 102.
29 IOBs 102 provide an interface with external circuitry.

30 Figure 2 is a schematic diagram of a conventional
31 circuit to implement one of IOBs 102 of FPLD 101. IOB 102
32 includes buffers 201-202, pull-up resistor 203, and I/O pad

1 204. This conventional pull-up circuit limits the
2 flexibility of a system designer by limiting the options
3 available for defining a given logic state on the output
4 pad.

5 Figure 3 is a schematic diagram of a Configurable
6 System on a Chip (CSoC) in accordance with an embodiment of
7 the present invention.

8 Figure 4 is a schematic diagram of CSL PIO 400 in
9 accordance with an embodiment of the present invention.
10 The I signal is the input data signal to the CSL from I/O
11 pad 417. The O signal is the output data signal from the
12 CSL to I/O pad 417. The E signal is the output enable
13 signal for PIO 400. The cf_pu signal is the configuration
14 pull-up signal and the cf_pd signal is the configuration
15 pull-down signal.

16 PIO 400 is user-enabled to provide a static weak pull-
17 up mode, a static weak pull-down mode, a weak signal
18 follower mode, or none of the above. To operate PIO 400 in
19 a static weak pull-up mode, configuration pull-up signal
20 cf_pu is a logic one, configuration pull-down signal cf_pd
21 is a logic zero, and output enable signal E is a logic
22 zero. Note that the logic low output enable signal E tri-
23 states output buffer 410, thereby preventing any transfer
24 of output data signal O. Under these conditions, a logic
25 zero is provided to the gate of n-channel transistor 414,
26 thereby turning off that transistor, and a logic zero is
27 provided to the gate of p-channel transistor 413, thereby
28 turning on that transistor. Thus, in a static weak pull-up
29 mode, I/O pad 417 is coupled to the voltage supply source
30 V_{cc} through resistor 415 (hence the "weak" pull-up).

31 To operate PIO 400 in a static weak pull-down mode,
32 configuration pull-up signal cf_pu is a logic zero,

1 configuration pull-down signal `cf_pd` is a logic one, and
2 output enable signal `E` is a logic zero. Under these
3 conditions, a logic one is provided to the gate of p-
4 channel transistor 413, thereby turning off that
5 transistor, and a logic one is provided to n-channel
6 transistor 414, thereby turning on that transistor. Thus,
7 in a static weak pull-down mode, I/O pad 417 is coupled to
8 ground through resistor 416 (hence the "weak" pull-down).

9 To operate PIO 400 in a weak follower mode, the
10 configuration pull-up `cf_pu` and configuration pull-down
11 `cf_pd` signals are logic ones and the output enable signal `E`
12 is a logic zero. Under these conditions, the logic value
13 of the signal at pad 417 determines the state of transistor
14 413-415. For example, if the signal on I/O pad 417 is a
15 logic zero, then the input data signal `I` is a logic zero,
16 which turns on p-channel transistor 414 and turns off n-
17 channel transistor 413. In this manner, PIO 400 will
18 continue to pull the voltage on I/O pad 417 to a logic
19 zero. In contrast, if the signal on I/O pad 417 is a logic
20 one, then input data signal `I` is a logic one which turns
21 off n-channel transistor 414 and turns on p-channel
22 transistor 413. In this manner, PIO 400 continues to pull
23 the voltage on I/O pad 417 to a logic one.

24 When the configuration pull-up `cf_pu` and configuration
25 pull-down `cf_pd` signals are logic zeros, PIO 400 does not
26 provide a pull-up or pull-down on I/O pad 417.

27 As described above, PIO 400 allows a user to
28 programmably enable a signal follower on each input data
29 signal present on I/O pad 417. Additionally, the signal
30 follower of PIO 400 avoids the power use, signal noise, and
31 time required to actively drive the data signal on I/O pad
32 417 to the level of a static weak pull-up or pull-down.

1 Therefore, maintenance of the data signal on I/O pad 417 at
2 the current state avoids many high current input
3 conditions.

4 Programmable CSL PIO 400 offers improved control over
5 the interactions on CSOC 300. For example, if I/O pad 417
6 provided an active high select signal to an external chip
7 on a board, then programming CSL PIO 400 to drive I/O pad
8 417 low in the absence of a select signal prevents spurious
9 selection of that external chip.

10 Figure 5 is a schematic diagram of CSL PIO 500 in
11 accordance with another embodiment of the present
12 invention. Similar elements between PIOs in Figures 4 and
13 5 are labeled similarly. PIO 500 includes buffers 410-411,
14 p-channel transistor 413, n-channel transistor 414,
15 resistors 415-416, I/O pad 417, user output enable logic
16 501, user output data logic 502, memory cells 503-508 and
17 multiplexers 509-512. The configuration signal CONFIG is a
18 logic one when the CSL is being configured and a logic zero
19 when the CSL is in normal operation.

20 PIO 500 also provides the user with the ability to
21 enable a static weak pull-up mode, a static weak pull-down
22 mode, or none of the above, as described above with respect
23 to Figure 4. However, PIO 500 does not provide a weak
24 follower mode. PIO 500 additionally provides the user with
25 the ability to define the output enable signal E and the
26 output data signal O as well as the ability to inject an
27 actively driven signal onto I/O pad 417 during
28 reconfiguration of the CSL. The operation of PIO 500 is
29 defined by the tables of Figures 5A and 5B.

30 Figure 5A describes the normal operation of PIO 500
31 when the configuration signal CONFIG is a logic zero. Note
32 that the software is set to disallow the state of memory

1 cell 506 storing a logic zero and memory cell 508 storing a
2 logic one. This logic zero of the configuration signal
3 CONFIG passes a user-defined signal from user output enable
4 logic 501 as the output enable signal E of multiplexer 509.
5 Additionally, the logic zero of the configuration signal
6 CONFIG passes a user-defined signal from user output data
7 logic 502 as the output data signal O of multiplexer 510.

8 The signal value at node A controls p-channel
9 transistor 413. During normal operation of the CSL, the
10 logic zero of the configuration signal CONFIG causes
11 multiplexer 511 to pass the value programmed in memory cell
12 506 as an output signal at node A. The signal value at
13 node B controls n-channel transistor 414. During normal
14 operation of the CSL, the logic zero of the configuration
15 signal CONFIG causes multiplexer 512 to pass the value
16 programmed in memory cell 508 as an output signal at node
17 B.

18 Logic zeros stored in memory cells 506 and 508 turn on
19 p-channel transistor 413 and turn off n-channel transistor
20 414. As a result, PIO 500 acts as a static weak pull-up
21 circuit when user output enable logic 501 is a logic zero.
22 However, PIO 500 drives I/O pad 417 to the value of user
23 output data logic 502 when user output enable logic 501 is
24 a logic one while still providing a weak pull-up.

25 A logic one stored in memory cell 506 turns off p-
26 channel transistor 413 and a logic zero stored in memory
27 cell 508 turns off n-channel transistor 414. As a result,
28 there is no mode active when user output enable logic 501
29 is a logic zero. However, PIO 500 drives I/O pad 417 to
30 the value of user output data logic 502 when user output
31 enable logic 501 is a logic one.

1 A logic one stored in memory cell 506 turns off p-
2 channel transistor 413 and a logic one stored in memory
3 cell 508 turns on n-channel transistor 414. As a result,
4 PIO 500 acts as a weak pull-down when user output enable
5 logic 501 is a logic zero. However, PIO 500 drives I/O pad
6 417 to the value of user output data logic 502 when user
7 output enable logic 501 is a logic one while still
8 providing a weak pull-down.

9 Figure 5B describes the configuration of PIO 500 when
10 the configuration signal CONFIG is a logic one. Note that
11 the software is set to disallow the state of memory cell
12 506 storing a logic zero and memory cell 508 storing a
13 logic one. Additionally, the software is set to disallow
14 memory cells 505 and 507 from having the same value when
15 the user output enable logic 501 is a logic one.

16 The logic one configuration signal CONFIG passes the
17 value stored in memory cell 503 as the output enable signal
18 E of multiplexer 509 and the value stored in memory cell
19 504 as the output data signal O of multiplexer 510.

20 The signal value at node A controls p-channel
21 transistor 413. During configuration of the CSL, the logic
22 one configuration signal CONFIG causes multiplexer 511 to
23 pass the value programmed in memory cell 505 as an output
24 signal at node A. The signal value at node B controls n-
25 channel transistor 414. During configuration of the CSL,
26 the logic one configuration signal CONFIG causes
27 multiplexer 512 to pass the value programmed in memory cell
28 507 as an output signal at node B.

29 A logic zero stored in memory cell 505 turns on p-
30 channel transistor 413 and a logic zero of memory cell 507
31 turns off n-channel transistor 414. As a result, PIO 500

1 acts as a static weak pull-up circuit when user output
2 enable logic 501 is a logic zero.

3 A logic one stored in memory cell 505 turns off p-
4 channel transistor 413 and a logic zero stored in memory
5 cell 507 turns off n-channel transistor 414. As a result,
6 there is no mode active when user output enable logic 501
7 is a logic zero. However, PIO 500 drives I/O pad 417
8 strongly to the value stored in memory cell 504 when user
9 output enable logic 501 is a logic one. A logic one stored
10 in memory cell 505 turns off p-channel transistor 413 and a
11 logic one stored in memory cell 507 turns on n-channel
12 transistor 414. As a result, PIO 500 acts as a weak pull-
13 down when user output enable logic 501 is a logic zero.

14 On power up of the CSoC, memory cells 503, 505 and 507
15 all store logic zeros to guarantee that I/O pad 417 is tri-
16 stated and that a weak pull-up mode is active. This
17 configuration beneficially allows modification early in the
18 configuration sequence. On subsequent re-configuration,
19 the voltage in I/O pad 417 is determined by the truth table
20 of Figure 5B. Note that during initial configuration,
21 other chips on a board are often reset. However, during
22 reconfiguration, other chips on the board are often
23 actively running.

24 Figure 6 is a schematic diagram of a programmable
25 delay circuit 600 in accordance with an embodiment of the
26 present invention. Programmable delay circuit 600 includes
27 input pads 601-602, clock path 613, data path 614 and input
28 register 607. Clock path 613 includes buffers 604-605 and
29 clock distribution system 621. Data path 614 includes
30 buffer 603, multiplexer 606, memory cells 608-611, and
31 input delay system 620. Input register 607 has a data

1 input terminal and a clock input terminal. Other
2 embodiments may have other numbers of memory cells.

3 Input pad 601 provides the Data_In signal to the data
4 input terminal of input register 607 via data path 614.
5 Input pad 602 provides the Global_Clock_In signal to the
6 clock input terminal of input register 607 via clock path
7 613.

8 Clock distribution system 621 contains an inherent
9 delay due to routing and buffering. For example, this
10 delay may come from a clock tree distribution of the
11 Global_Clock_In signal. This inherent delay means data
12 present at the data input terminal of input register 607
13 must "wait" for the clock signal to arrive. Thus, the data
14 present at the data input terminal can not be clocked into
15 input register 607 until the delayed clock signal arrives
16 at the clock input terminal. This amount of time that data
17 has to wait is called the "hold time" of the circuit.

18 The hold time of the circuit can be minimized by
19 adding delay to data path 614 of the circuit. If an amount
20 of delay equivalent to the amount of delay in clock
21 distribution system 621 can be added to data path 614, the
22 hold time of the circuit can be lessened to substantially
23 zero. Thus, a circuit has "zero hold time" when the delay
24 in data path 614 equals the delay in clock path 613. When
25 a circuit has zero hold time, the data signal does not need
26 to wait at the data input terminal. Therefore, the data
27 signal may change at substantially the same time as the
28 clock signal.

29 Multiplexer 606 provides the user with the option to
30 use the Data_In signal or the delayed Data_In signal, based
31 on the value stored in memory cell 611. Input delay system
32 620 uses the values stored in memory cells 608-611 to

1 select one of eight different delay paths. Thus, the user
2 is able to select the amount of delay in programmable delay
3 circuit 600. This programmability of the data path delay
4 is especially time-saving in situations where the
5 manufacturing process is not well-characterized and the
6 clock distribution delay is not well known. In the present
7 invention, the user only has to design the range of delays
8 available in input delay system 620 to cover any possible
9 delay in clock path 613. Input delay system 620 is shown
10 in greater detail in Figure 7.

11 Another use for the programmability of input delay
12 system 620 is to allow the user to tailor the data delay
13 after the place and route of the design is finished. At
14 that point, the clock delay of clock path 613 may be
15 characterized to optimize system performance.

16 Figure 7 describes one embodiment of the input delay
17 system 620 of programmable delay circuit 600 (Figure 6).
18 Input delay system 620 includes p-channel transistors 701-
19 716, n-channel transistors 717-726 and 731-736, and
20 inverters 740-741. Data is provided to input delay system
21 620 at an input 742. Data is provided to the external
22 system from input delay system 620 at an output 743.

23 Memory cells 608-610 are used to select one of eight
24 possible combinations of transistor pairs. These
25 combinations of transistor pairs provide a resistive path
26 to charge and discharge the MOS capacitors formed by
27 transistors 709-710 and 719-720. For example, when memory
28 cells 608-610 each store the logic value "0", p-channel
29 transistors 711-716 (receiving logic zeros) and n-channel
30 transistors 731-736 (receiving logic ones) are all on. As
31 a result, the effective resistance provided by transistors
32 is at the smallest value. Transistors 707-708 and 717-718

1 are sized so that their effective resistance is changed by
2 turning on or off a series of different sized transistors.

3 P-channel transistors 701-703 are fabricated such that
4 they have resistance values in the proportion R , $2R$, and
5 $4R$, respectively. Therefore, p-channel transistor 703 has
6 twice the resistance value of p-channel transistor 702, and
7 p-channel transistor 702 has twice the resistance value of
8 p-channel transistor 701. Similarly, p-channel transistors
9 704-706 and n-channel transistors 721-723 and 724-726 have
10 resistance values in the proportion R , $2R$, and $4R$,
11 respectively. P-channel transistors 711-716 and n-channel
12 transistors 731-736 are fabricated to have resistance
13 values much less than R . Of course, other embodiments of
14 the present invention may have other proportions of
15 resistance values.

16 P-channel transistors 701-703 and 711-713 are coupled
17 to form a first series resistor circuit between the voltage
18 supply source V_{cc} and the source of p-channel transistor
19 707. N-channel transistors 721-723 and 731-733 are coupled
20 to form a second series resistor circuit between the drain
21 of n-channel transistor 717 and ground. P-channel
22 transistor 707 and n-channel transistor 717 are coupled to
23 form a first inverter with an input terminal coupled to the
24 output of inverter 740 and an output terminal at node $N1$.
25 This first inverter serves to couple node $N1$ to either the
26 voltage supply source V_{cc} through the first series resistor
27 circuit or ground through the second series resistor
28 circuit. A logic zero on the input terminal of the first
29 inverter couples the first series resistor circuit to node
30 $N1$, thereby providing a current source to node $N1$. A logic
31 one on the input terminal of the first inverter couples the

1 second series resistor circuit to node N1, thereby
2 providing a current sink from node N1.

3 The logic values stored within memory cells 608-610
4 are coupled to the gates of p-channel transistors 711-713.
5 When memory cells 608-610 all store logic values logic one,
6 each of p-channel transistors 711-713 is turned off. As a
7 result, the resistance values of p-channel transistors 701-
8 703 form the resistance of the first series resistor
9 circuit. Therefore, the resistance of the first series
10 resistor circuit formed by p-channel transistors 701-703
11 and 711-713 has a maximum resistance value of $7R$.
12 Similarly, these logic values of memory cells 608-610 turn
13 off each of n-channel transistors 731-733. As a result,
14 the resistance values of n-channel transistors 721-723 form
15 the resistance of the second series resistor circuit.
16 Therefore, the resistance of the second series resistor
17 circuit formed by n-channel transistors 721-723 and 731-733
18 has a maximum resistance value of $7R$.

19 In this situation, where memory cells 608-610 all
20 store a logic one, both the first and the second series
21 resistor circuits have resistance values of $7R$. Therefore,
22 a relatively small current will flow either from the first
23 series resistor circuit to node N1 or from node N1 to the
24 second series resistor circuit.

25 P-channel transistor 709 has both a source and a drain
26 coupled to voltage supply source V_{cc} . N-channel transistor
27 719 has both a source and a drain coupled to ground. P-
28 channel transistor 709 and n-channel transistor 719 each
29 have a gate coupled to node N1, thereby forming two
30 capacitors.

31 When the data signal at input 742 is a logic zero, the
32 voltage at node N1 is pulled down to a logic zero through

1 the second series resistor circuit formed by n-channel
2 transistors 721-723. This logic zero at node N1 turns on
3 p-channel transistor 709 and turns off n-channel transistor
4 719.

5 When the data signal at input 742 transitions to a
6 logic one, the output voltage of the first inverter formed
7 by p-channel transistor 707 and n-channel transistor 717
8 transitions from a logic zero to a logic one. When the
9 voltage at node N1 reaches a voltage equal to one threshold
10 voltage, n-channel transistor 719 turns on lightly. At
11 this time, a gate capacitance appears at node N1. This
12 gate capacitance causes the relatively small current to
13 take a long time to charge up node N1 to a logic one,
14 thereby delaying the input signal at input 742.

15 P-channel transistors 704-706 and 714-716 form a third
16 series resistor circuit similar to the first series
17 resistor circuit. N-channel transistors 724-726 and 734-
18 736 form a series resistor circuit similar to the second
19 series resistor circuit. P-channel transistor 710 and n-
20 channel transistor 720 form a capacitor pair similar to p-
21 channel transistor 709 and n-channel transistor 719.
22 Therefore, in a similar fashion to that described above,
23 the input signal at input 742 is further delayed through
24 input delay system 620. The series resistor circuits in
25 combination with capacitors 709-710 and 719-720 form a
26 series of two resistor-capacitor (RC) segments.

27 When memory cells 608 and 610 both store logic zeros
28 and memory cell 609 stores a logic one, p-channel
29 transistors 711 and 713 are turned on, while p-channel
30 transistor 712 is turned off. P-channel transistors 711
31 and 713 short transistors 701 and 703, respectively,
32 thereby lessening the resistance value of the first series

1 resistor circuit to $2R$. Similarly, N-channel transistors
2 731 and 733 short n-channel transistors 721 and 723,
3 respectively, thereby lessening the resistance value of the
4 second series resistor circuit to $2R$. As a result, the
5 current flowing through node N1 is greater under these
6 circumstances than the relatively small current flowing
7 through node N1 when memory cells 608-610 all stored logic
8 ones.

9 The effect of this larger current flowing through node
10 N1 is to take less time to charge the capacitors formed by
11 p-channel transistor 709 and n-channel transistor 719.
12 Similarly, the larger current flowing through node N2
13 allows the capacitors formed by p-channel transistor 710
14 and n-channel transistor 720 to charge more quickly. As a
15 result, the delay in passing the data value from input 742
16 through input delay system 620 is much shorter than
17 described above.

18 When memory cells 608-610 all store logic ones, they
19 provide a large resistance in a series resistor circuit.
20 This large resistance results in a small current. The
21 small current takes a relatively long time to charge
22 capacitors, thereby causing a relatively long delay within
23 input delay system 620. When memory cells 608-610 all
24 store logic zeros, they provide a small resistance in a
25 series resistor circuit. This small resistance results in
26 a large current. The large current takes a relatively
27 short time to charge capacitors, thereby causing a
28 relatively short delay within input delay system 620. In
29 this manner, the delay within input delay system 620 is
30 programmable by the user.

1 Inverters 740-741 are used to buffer the data input
2 and data output of delay circuit element 620, allowing this
3 circuit to stand alone.

4 Figure 8A is a schematic diagram of placement-
5 independent edge tiles in a semi-homogeneous logic array in
6 accordance with an embodiment of the present invention.

7 Array of tiles 800 is part of a homogeneous array of
8 larger tiles forming a programmable logic architecture.
9 Array of tiles 800 comprises interior logic block tiles
10 801-802 and identical edge-placed I/O tiles 850-859.
11 Identical edge-placed I/O tiles 850-859 are designed such
12 that the external signals they receive are similarly or
13 identically available to interior logic block tiles 801-802
14 without regard to the edge at which I/O tiles 850-859 are
15 placed.

16 I/O tile 850 includes vertical input multiplexer 810,
17 horizontal input multiplexer 820, OR gate 830 and internal
18 circuitry 840. Identical I/O tiles 851-859 include
19 vertical input multiplexers 811-819, horizontal input
20 multiplexers 821-829, OR gates 831-839, and internal
21 circuitry 841-849, respectively.

22 I/O tiles 850-859 are designed to have the same
23 physical dimensions as the tiles to which they abut. All
24 general CSL interconnect and power routing along the edge
25 of I/O tiles 850-859 must conform with the dimensions of
26 abutting tiles on all sides. Routing channels which carry
27 signals not generated within, nor needed to pass through,
28 array of tiles 800 may be used to convey external I/O
29 signals. This connectivity may be defined by simple metal-
30 layer programming during layout.

31 The number of input multiplexers (e.g., 810 and 820)
32 which bring general CSL interconnect lines into the

1 internal circuits of the I/O tile is doubled from
2 conventional methods. Conventional IOBs use a single
3 multiplexer coupled to receive twice the number of input
4 signals of input multiplexer 810. This single multiplexer
5 is expensive in terms of time required to test the I/O tile
6 during production. The use of two multiplexers each
7 receiving half the number of signal of the conventional
8 multiplexer halves the number of test configurations
9 required during production. Note that it is not necessary
10 to double the number of, for example, horizontal-channel
11 input multiplexers to guarantee that a horizontally run
12 signal is available to the same input in an I/O tile
13 regardless of the placement of the tile. Each horizontal-
14 channel input multiplexer output is ORed with a vertical
15 channel input multiplexer output. Efficiency may be
16 additionally increased by using fewer input multiplexers if
17 the input signals are swappable.

18 Particular benefits of the I/O tiles 850-859 of the
19 present invention include the ability to logically reside
20 along any external edge due to the ORing of horizontal and
21 vertical input multiplexer inputs. Additionally, ensuring
22 identical dimensions and routing allows I/O tiles 850-859
23 to physically reside along any external edge. Furthermore,
24 the I/O tiles 850-859 may be coupled with additional tiles
25 because of their physical similarity to interior logic
26 block tiles. Lastly, layout and design verification is
27 greatly simplified with the present I/O tiles 850-859.

28 Figure 8b is a detailed schematic diagram of an
29 identical edge-placed I/O tile 850 having input
30 multiplexers which provide the value of the output data O
31 and the output enable E signals to PIO 400 (Figure 4).

1 Input multiplexer 820 receives a logic zero at an
2 first input terminal, a logic one at a second input
3 terminal, and sixteen (16) signals 804 from vertical
4 routing channel at sixteen other input terminals. Input
5 multiplexer 820 receives control signals from a plurality
6 of memory cells 806. A vertical routing channel is a
7 routing channel within the general CSL interconnect. Input
8 multiplexer 810 receives a logic zero at an first input
9 terminal, a logic one at a second input terminal, and
10 sixteen (16) signals from horizontal routing channel 803 at
11 sixteen other input terminals. Input multiplexer 810
12 receives control signals from a plurality of memory cells
13 805. A horizontal routing channel is a routing channel
14 within the general CSL interconnect. Output data signal O
15 to PIO 400 is the logical OR of the output signal of input
16 multiplexers 820 and 810. Similarly, the output enable
17 signal E to PIO 400 is the logical OR of the output signal
18 of input multiplexers 820 and 810. By having the option to
19 select signals from both horizontal routing channel 803 and
20 vertical routing channel 804, multiple PIOs similar to PIO
21 400 can be used in different locations. Therefore, the
22 same PIO design layout can be used on all four edges of the
23 chip containing the CSL.

24 The use of the input multiplexers providing output
25 data O and output enable E signals to PIO 400 provide
26 enhanced testability for PIO 400. Specifically, the use of
27 OR gate 830 allows one horizontal signal of horizontal
28 routing channel 803 and one vertical signal of vertical
29 routing channel 804 to be tested in the same test
30 configuration. As a result, the single test configuration
31 can test both horizontal and vertical connections by
32 alternately forcing the vertical and the horizontal signals

1 to a logic zero. Signals are forced to zero by routing a
2 signal from one of the tester controllable resources (not
3 shown) to the applicable input multiplexer. Therefore, to
4 test all thirty-six configurations (signals from both
5 horizontal routing channel 803 and vertical routing channel
6 804), only 18 test configurations are needed. Note that
7 conventional implementation requires the use of one thirty-
8 six by 1 input multiplexer to implement the same test
9 configuration. Therefore, a conventional test of these 36
10 signals requires thirty-six test configurations, which is
11 double the number required by the present invention.

12 Figure 9 is a schematic diagram of fast switches in
13 accordance with an embodiment of the present invention.

14 N-channel transistor switches 920-921 are coupled
15 between pairs of I/O pads. I/O pad 910 is conventionally
16 coupled to provide an input data I signal to user logic
17 930, and to receive an output data signal O through buffer
18 905, if enabled by output enable signal E. Similarly, I/O
19 pads 911-913 are conventionally coupled to provide an input
20 data I signal to user logic 931-933, respectively, and
21 receive an output data signal O through buffers 906-908,
22 respectively, if enabled by the output enable signal E.

23 User logic 934 provides a control signal to buffer
24 908. N-channel transistor switches 920-921 each have a
25 gate coupled to the output signal of buffer 908.
26 Therefore, a logic one asserted by user logic 934 turns on
27 n-channel transistor switches 920-921. Turned on
28 transistor switch 920 couples I/O pad 910 to I/O pad 912.
29 As a result, the signal from I/O pad 912 can be transferred
30 to I/O pad 910 without having to drive the input data I
31 signal from user logic 932 through routing to the output
32 data O signal of user logic 930. Therefore, I/O pads 910

1 and 912 are connected together with minimal propagation
2 delay. Turned on transistor switch 921 couples I/O pad 913
3 to I/O pad 911 in a similar manner.

4 Switches 920-921 can be implemented in CMOS
5 transmission gates. Alternatively, switches 920-921 can be
6 implemented in NMOS pass transistors if the threshold drop
7 while passing a logic one can be tolerated. Switches 920-
8 921 can also be implemented in NMOS pass transistors having
9 gates boosted by any of various conventional circuit
10 techniques.

11 Figure 10 is a schematic diagram of a latch bypass
12 1000 in accordance with an embodiment of the present
13 invention.

14 Latch bypass 1000 includes OR gate 1001, transistor
15 1002 and latch 1005. Latch 1005 includes inverters 1003-
16 1004. A first route within routing 1007, Route_A, receives
17 the data input signal through latch 1005. Routing 1007 is
18 part of the general CSL interconnect of the CSoC. A second
19 route within routing 1007, Route_B, receives the data input
20 signal directly.

21 An input data signal I is provided by PIO 400 (Fig. 4)
22 to the source of transistor 1002. As noted above, this
23 input data signal I is also provided by PIO 400 directly to
24 Route_B within routing 1007. The bypass signal provided by
25 memory cell 1006 to the first input terminal of OR gate
26 1001 is generated by a programmable memory element in the
27 CSL. A logic one bypass signal forces latch bypass 1000
28 into bypass mode. When latch bypass 1000 is in bypass
29 mode, the current value of the input data signal I is
30 available to routing 1007 via Route_A.

31 When latch bypass 1000 is in bypass mode, the output
32 signal provided to Route_A is allowed to directly follow

1 the input data signal I. A logic one output enable signal
2 E or a logic one bypass signal from PIO 400 turns on
3 transistor 1002. As a result, the input data signal I is
4 coupled to the input node of latch 1005 through turned on
5 transistor 1002. Therefore, both Route_A and Route_B
6 within routing 1007 receive the current logic value of the
7 input data signal, I.

8 When both the output enable signal E and the bypass
9 signal are logic zeros, transistor 1002 is turned off.
10 Under these circumstances, storage latch 1005 latches the
11 value of the input data signal I. Therefore, Route_A
12 within routing 1007 receives the latched logic value of the
13 input data signal, I. Thus, the state of the input data
14 signal I is preserved. If the input data signal I changes
15 state, the current value of the input data signal I is
16 available to routing 1007 through Route_B and the latched
17 value of the input data signal is available to routing 1007
18 through Route_B.

19 The bypass signal may be permanently enabled, so that
20 the input data signal I is always available to both Route_A
21 and Route_B of routing 1007. As a result, the number of
22 channels available for a direct connection to the input
23 data signal I is increased. The availability latch bypass
24 1000 to provide both the current and previous input data
25 signal I to the general CSL interconnect 1007 supplants the
26 need for a multiplexer to receive both current and previous
27 input data signals. Therefore, this implementation of
28 latch bypass 1000 requires fewer multiplexers on the CSoc,
29 thereby decreasing circuit area and decreasing additional
30 delay involved in signal selection.

31 Although the present invention has been described in
32 connection with one embodiment, it is understood that this

1 invention is not limited to such embodiment, but is capable
2 of various modifications which would be apparent to a
3 person skilled in the art. Thus, the invention is limited
4 only by the following claims.